

The Database Update

There have been a few significant changes made to the project in this version that will allow the application to retrieve its data from a local database. The UI remains unchanged for this update, but there are design ideas that are likely to be applied soon. The stack being used in this project is MySQL, Express, and Angular. The reason why we decided on this stack is because MySQL is what we are most familiar with, and it allows for easy local connections. Express was a natural choice as we were already using Node for Angular and have worked with it in previous courses.

Once I created the database and table, I ran the MySQL server on the default **port 3306**.

The express **app.js** file then retrieves information from there and sends it in a Json format on **port 3000**.

In the typescript file of the component that I want to retrieve data, I include...

```
export class PageSplashComponent implements OnInit {
  // Data from JSON file
  data: any = employeeData;

  // Array of employees (from JSON)
  employees: any[] = this.data.employees;

  apiData: any;

  constructor(private http: HttpClient) {}

  ngOnInit(): void {
    this.getDataFromServer();
  }

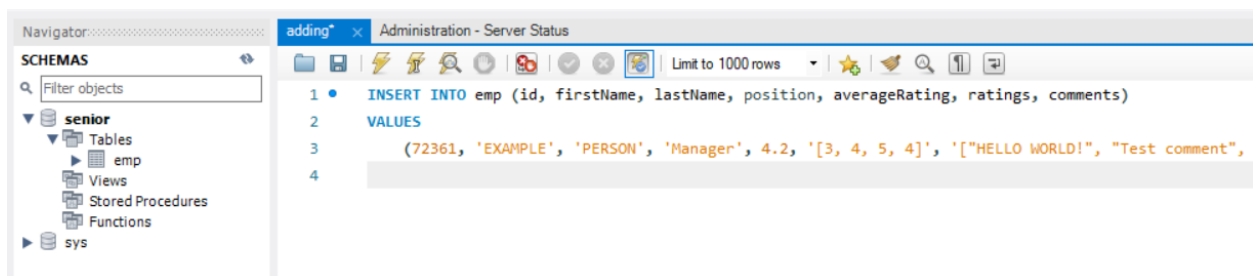
  getDataFromServer() {
    this.http.get('http://localhost:3000/api/data').subscribe(
      (data: any) => {
        console.log('Data from server:', data);
        this.apiData = data;
      },
      (error) => {
        console.error('Error fetching data from server:', error);
      }
    );
  }
}
```

This gets the information from the API and backend and stores it within the **apiData** variable. This can then be used identically to the dummy Json that was included in prior builds.

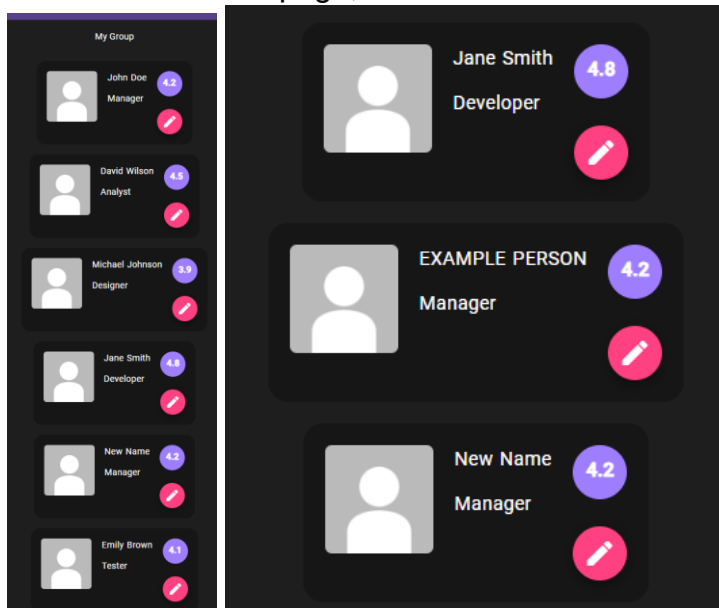
As such, I have updated the variable that the for loop pulls from on the frontend-

```
<app-splashblock
  *ngFor="let i of apiData"
  fName="{{ i.firstName }}"
  lName="{{ i.lastName }}"
  pos="{{ i.position }}"
  rating="{{ i.averageRating }}"
  id="{{ i.id }}"
  [comments]="i.comments"
></app-splashblock>
```

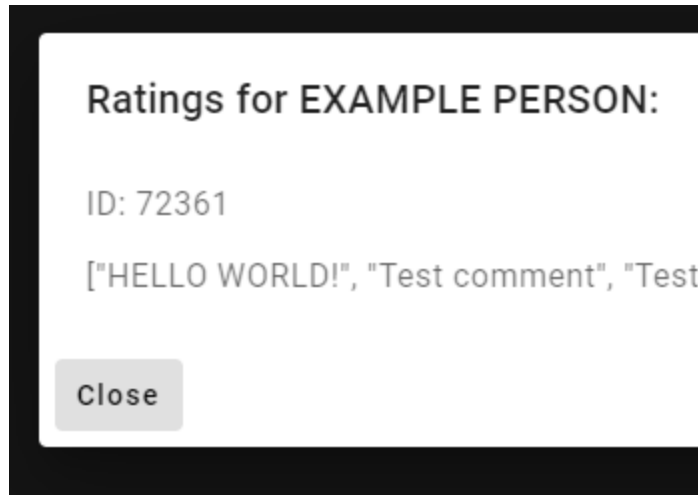
I can now enter new values into the database table using the MySQL workbench:



When I refresh the page, the list now includes this new entry:



Clicking on the employee profile to access the comments reflects these changes as well:



In GitHub, I created a separate branch for working with the local server running, which will be used to test backend work. For most of the frontend work, we will be using the main development branch and retrieving data from the dummy Json file.